链滴

分享一个 web 应用版本监测 (更新) 的工具库

作者：akimyou

原文链接：https://ld246.com/article/1659422128955

来源网站：链滴

bell version-rocket rocket

简体中文 | English

一个用于 web 应用检测版本更新的小工具。

经常会发生这样的情况: 当用户在浏览器中打开某 web 应用较长时间且未刷新页面, 在应用有新版本新或问题修复时, 用户会无法及时知晓有新版发布, 导致用户继续使用旧的版本, 影响用户体验和后端据准确性。

在团队合作中可能会有这样的情况: 你作为前端工程师, 在联调测试或部署上线时, 每次部署后都需要团队成员口头传达已经部署成功, 增加了沟通成本, 不够自动化, 也没有部署记录以有迹可循。

使用 **version-rocket** 可以帮你解决以上困扰。

简介

version-rocket 将用户当前浏览器中的版本与远程服务器中的版本文件进行比较。

如果有新的版本发布，将在页面中展示一个新版本更新提示弹窗，用户可以通过点击刷新按钮来更新本。另外，**version-rocket** 支持个性化设置提示弹窗文案和主题, 也可传入一个回调函数来自定义版更新提示。

我们使用基于 javascript 的 **Web Worker API** 来做监测轮询，不会影响浏览器渲染进程。

另外, 如果你所在的团队, 使用 Lark 或 飞书来团队协作, **version-rocket** 可以帮你推送 “部署成功” 消息到 Lark 群聊中 (通过 Lark 机器人)。 使用方法非常快捷简单, 使用方法见下文。

如果有其他平台的推送需求, 可以提 issue

功能特点

- 支持所有现代浏览器
- 可用版本实时监测, 支持任意版本格式, 例如: 1.1.0、1.1.1.0、1.1.0-beta 等等
- 支持个性化设置版本提示弹窗的文案和主题, 也支持自定义 UI
- 部署成功后，将部署消息同步到 Lark 群聊
- 部署信息卡片的文案和消息模版支持自定义, 并支持动态生成的字段传入
- 支持 TypeScript
- 支持 Npm 安装

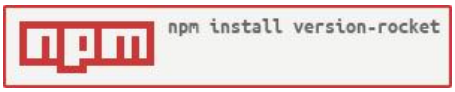
效果截图

- **第一张图:** 当有新版本更新时, 及时提醒用户刷新页面的功能弹窗 (默认 UI)。
- **第二张图:** 个性化设置弹窗文案和主题, 有文案和主题有自定义需求时, 非常好用。

- **第三张图:** 在项目成功部署后，部署信息将被发送到群聊，以通知团队成员, 卡片文案通过一个 json 文件来配置, 使用方法请参见下文。
- **第四张图:** 自定义消息卡片文案, 可设置是否 @全员, 并支持动态生成的字段传入 (如 version 在 ci/c 后生成, 支持动态传入)

使用方法

安装



开始使用

安装最新版, 使用 checkVersion 方法, 该方法兼容 pollingCompareVersion 功能, 并且支持自定义窗文案和主题 (推荐)

使用默认主题

```
// 1. 第一步: 导入 checkVersion(), 并调用
import { checkVersion } from 'version-rocket'
// 默认建议使用 package.json 中的 version 字段, 若有自定义 version, 请忽略此行
import { version } from '../package.json'

checkVersion({
  localPackageVersion: version,
  originVersionFileUrl: `${location.origin}/version.json`,
})



/**
 * 2. 第二步
 * 执行 generate-version-file 快捷命令，在项目中生成 version.json 文件, 用于部署到远程服务器
 *
 * VERSION(可选): 默认使用 package.json 中的 version 生成 version.json 文件, 如需要自定义 vers
on, 可传入环境变量 VERSION 来定义
 *
 * version.json 文件默认生成在 dist 目录下, 如果需要自定义目录, 可传入目录参数, 参见以下示例:
*/

{
  "name": "test",
  "description": "test",
  "private": true,
  "version": "0.0.1",
  "scripts": {
    ...
```

```
    "generate:version": "VERSION=1.1.0-beta generate-version-file dist public"
    ...
  },
  ...
}
```

完成以上两个步骤, 版本监测功能已经就正常使用了
ada□tada

个性化设置弹窗文案和主题

```
import { checkVersion } from 'version-rocket'
import { version } from '../package.json'

checkVersion(
  {
    localPackageVersion: version,
    originVersionFileUrl: `${location.origin}/version.json`,
  },
  {
    title: 'Title',
    description: 'Description',
    primaryColor: '#758bfd',
    rocketColor: '#ff8600',
    buttonText: 'Button Text',
  }
)
```

个性化设置弹窗提示图片

```
import { checkVersion } from 'version-rocket'
import { version } from '../package.json'

checkVersion(
  {
    localPackageVersion: version,
    originVersionFileUrl: `${location.origin}/version.json`,
  },
  {
    imageUrl: 'https://avatars.githubusercontent.com/u/26329117',
  }
)
```

如果在使用 version 1.0.9 及以下版本, 调用 pollingCompareVersion 方法

推荐升级为最新版, 体验自定义弹窗主体和文案的功能

```
// 1. 第一步: 导入 pollingCompareVersion 方法并调用
import { pollingCompareVersion } from 'version-rocket'
import { version } from '../package.json'

pollingCompareVersion(version, `${location.origin}/version.json`, 30000, (data) => {
    console.log(data)
})

// 2. 第二步: 请参见上文: “使用默认主题”
```

更多个性化设置参见“属性/参数”表 page_facing_up

支持推送部署成功的通知到 Lark 群聊

```
/**
 * 1. 第一步:
 * 你需要在项目根目录下创建一个 send-lark-config.json 文件，它存储了用于设置消息卡片展示文的字段
 *
 * 然后, 执行 send-lark-message 快捷命令。默认情况下，当前路径中的 send-lark-config.json 文被选中
 *
 * MESSAGE_PATH (可选): 如果你想自定义文件路径或文件名，你可以设置 MESSAGE_PATH 参数将其传入 (此参数对有区分部署环境的需求时, 非常有用)
 *
 * PACKAGE_JSON_PATH (可选): 如果你需要自定义 package.json 文件路径, 可以设置 PACKAGE_JON_PATH 参数来自定义 (此参数对于 monorepo 项目的部署时, 可能有用。不传此参数, 默认获取根径下的 package.json 文件)
*/

{
  "name": "test",
  "description": "test",
  "private": true,
  "version": "0.0.1",
  "scripts": {
    ...
    "send-lark-message:test": "MESSAGE_PATH=./lark-message-staging-config.json PACKAGE_SON_PATH=./packages/test/package.json send-lark-message"
    ...
  },
  ...
}



// 第二步: 配置 send-lark-config.json 文件

{
```

```
    // 消息卡片标题
    "title": "TEST FE Deployed Successfully",
    // 项目名称标签
    "projectNameLabel": "Project name label",
    // 项目名称
    "projectName": "TEST",
    // 项目分支标签
    "branchLabel": "Branch label",
    // 项目分支, 可用于区别部署环境
    "branch": "Staging",
    // 版本标签
    "versionLabel": "Version label",
    // 版本
    "version": "1.1.1.0",
    // 项目可访问地址标签
    "accessUrlLabel": "Access URL label",
    // 项目可访问地址
    "accessUrl": "https://test.com",
    // 是否@所有人标签
    "isNotifyAllLabel": "Is notify all label",
    // 是否@所有人: true / false
    "isNotifyAll": true,
    // Lark 机器人的 webhook 链接
    "larkWebHook": "https://open.larksuite.com/open-apis/bot/v2/hook/xxxxxxxxxxxx",
    // 可选: 部署工具描述
    "deployToolsText": "Deploy tools text",
    // 可选: 部署所使用的方式或平台
    "deployTools": "Jenkins",
    // 可选: 部署时间想要转换成的时区，默认 "Asia/Shanghai" (当你的项目要部署的目标服务器与所在时区不同, 可以设置此字段来转换时区)
    "expectConvertToTimezone": "America/New_York"
    // 可选: 想要展示除模版之外的更多信息
    "remark": "Trigger by bob, fix xxx bug"
}
```

支持传入动态生成的卡片文案

当你的卡片文案会根据条件动态生成时, 可以传入 MESSAGE_JSON 字段来定义, 注意: MESSAGE_JSON 的值需要做转义

```
  /**
   * MESSAGE_JSON (可选): 如果你的卡片文案会根据条件来生成, 可以传入 MESSAGE_JSON 字段自定义 (此参数对于卡片文案动态生成的应用, 非常有用, 如 version, title 等)
  */

  {
    "name": "test",
    "description": "test",
    "private": true,
    "version": "0.0.1",
    "scripts": {
      ...
```

```
    "send-lark-message:test": "MESSAGE_JSON='{\"title\":\"This is a dynamically generated tit
e\",\"version\":\"1.1.0-beta\",\"accessUrl\":\"http://test.example.com\",\"isNotifyAll\":true}' se
d-lark-message"
      ...
    },
    ...
  }



// 或者 export 变量后, 在 package.json 中引用

// ci file
sh "npm run build"
sh "export messageJSON='{\"title\": \"This is a title\"}'"

// package.json
{
    "name": "test",
    "description": "test",
    "private": true,
    "version": "0.0.1",
    "scripts": {
      ...
      "send-lark-message:test": "MESSAGE_JSON=${messageJSON} send-lark-message"
      ...
    },
    ...
  }
```

个性化设置部署消息卡片

```
// send-lark-config.json 示例

{
    // 消息卡片内容
    "message": {
        "msg_type": "text",
        "content": {
            "text": "New message reminder"
        }
    },
    // Lark 机器人的 webhook 链接
    "larkWebHook": "https://open.larksuite.com/open-apis/bot/v2/hook/xxxxxxxxxxxx"
}
```

属性/参数

checkVersion 函数参数表

参数	类型	描述	默认值	必需
config	object	版本监测配置项	无	是
config.originVersionFileUrl	string	远程服务器上的 version.json 文件路径	无	是
config.localPackageVersion	string	当前应用版本号, 通常取 package.json 的 version 字段, 用于与远程服务器的 version.json 文件比较	无	是
config.pollingTime	number	轮询监测的时间间隔, 单位 ms	1000	否
config.onVersionUpdate	function(data)	自定义版本提示 UI 的回调函数 (如果你想自定义弹窗 UI, 通过回调函数可以拿到返回值来控制弹窗的显隐)	无	否
options	object	弹窗文案和主题的配置项 (不自定义弹窗 UI, 但有修改文案和主题的需求时使用)	无	否
options.title	string	弹窗的标题	Update	否
options.description	string	弹窗的描述	xxx is available	否
options.buttonText	string	弹窗按钮的文案	efresh	否
options.imageUrl	string	弹窗的提示图片		否
options.rocketColor	string	弹窗提示图片中火箭的主题色, 置后 options.imageUrl 无效		否
options.primaryColor	string	弹窗的主题色, 会作用到提示片背景色和按钮背景色, 设置后 imageUrl 无效		否
options.buttonStyle	string	弹窗按钮的 css 配置, 可以覆盖默认的按钮样式	无	否

pollingCompareVersion 函数参数表

参数	类型	描述	默认值	必需
localPackageVersion	string	当前应用版本号, 通常取 package.json 的 version 字段, 用于与远程服务器的 version.json 文件比较	无	是
originVersionFileUrl	string	远程服务器上的 version.json 件路径	无	是
pollingTime	number	轮询监测的时间间隔, 单位 ms		
onVersionUpdate	function(data)	自定义版本提示 UI 的回函数 (如果你想自定义弹窗 UI, 通过回调函数可以拿到返回值来控制弹窗的显隐)	无	

链接

- 时区参照表

- JSON 在线转义工具

许可证

version-rocket 是开源软件, 许可证为 Apache License 2.0